Subspace maximum margin clustering

Conference Paper · November 2009

DOI: 10.1145/1645953.1646122 · Source: DBLP

2 authors, including:

Quanquan Gu
University of Virginia
48 PUBLICATIONS **752** CITATIONS

Subspace Maximum Margin Clustering

Quanquan Gu
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing, China, 100084
gqq03@mails.tsinghua.edu.cn

Jie Zhou
State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing, China, 100084
jzhou@tsinghua.edu.cn

ABSTRACT

In text mining, we are often confronted with very high dimensional data. Clustering with high dimensional data is a challenging problem due to the *curse of dimensionality*. In this paper, to address this problem, we propose an subspace maximum margin clustering (SMMC) method, which performs dimensionality reduction and maximum margin clustering simultaneously within a unified framework. We aim to learn a subspace, in which we try to find a cluster assignment of the data points, together with a hyperplane classifier, such that the resultant margin is maximized among all possible cluster assignments and all possible subspaces. The original problem is transformed from learning the subspace to learning a positive semi-definite matrix, in order to avoid tuning the dimensionality of the subspace. The transformed problem can be solved efficiently via cutting plane technique and constrained concave-convex procedure (CCCP). Since the sub-problem in each iteration of CCCP is joint convex, alternating minimization is adopted to obtain the global optimum. Experiments on benchmark data sets illustrate that the proposed method outperforms the state of the art clustering methods as well as many dimensionality reduction based clustering approaches.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Experimentations

Keywords

Maximum margin clustering, Dimensionality reduction, Cutting plane, Constrained concave-convex procedure

CIKM'09, November 2–6, 2009, Hong Kong, China.

1. INTRODUCTION

Clustering has long been an important topic in machine learning and data mining. It aims to divide the unlabeled data set into groups of similar data points. Many clustering methods have been proposed up to now, e.g. K-means, spectral clustering [19] [17] and maximum margin clustering [22]. In text mining, we are often confronted with very high dimensional data. Clustering with the high dimensional data is a challenging problem due to the *curse of dimensionality*. Many dimensions in the data are redundant or irrelevant for clustering, so the clustering performance is usually affected. On the other hand, handling the high dimensional data usually leads to high computational cost.

To tackle this problem, dimensionality reduction is a potential approach since it may find a subspace which has more separability and discriminative information for clustering. In the past decades, many dimensionality reduction based clustering methods have been proposed [7] [15] [9] [14] [8] [24] [25], which do dimensionality reduction and k-means simultaneously.

In this paper, we present an subspace maximum margin clustering (SMMC), which integrates dimensionality reduction with the state of the art clustering method, i.e. maximum margin clustering [22] in a joint framework. SMMC aims to learn a subspace, in which it tries to find a cluster assignment of the data points, together with a hyperplane classifier, such that the resultant margin is maximized among all possible cluster assignments and all possible subspaces. Since the dimensionality of the subspace is also a parameter which needs careful tuning, we transform the original problem from learning the subspace to learning a positive semi-definite matrix in order to avoid the difficulty of tuning it. Then the problem can be solved efficiently by cutting plane algorithm [13] and constrained concave-convex programming (CCCP) [21], which owns well-studied convergence properties. Furthermore, we will show that in each iteration of CCCP, the sub-problem is joint convex on the variables, hence alternating minimization can be adopted to obtain the global optimum. The convergence and the computational complexity of the whole algorithm is also analyzed. Experimental results on benchmark data sets illustrate that the proposed method outperforms existing dimensionality reduction based methods as well as state of the art clustering approaches.

The remainder of this paper is organized as follows. In Section 2, we will review some existing works related with ours. In Section 3, we will propose binary class subspace

maximum margin clustering. In Section 4, we will extend binary class subspace maximum margin clustering to multi-class setting. Experiments on benchmark data sets are demonstrated in Section 5. Finally, we draw a conclusion in Section 6.

2. RELATED WORKS

In this section, we will review some works closely related with ours. We first introduce dimensionality reduction based clustering approaches, followed with maximum margin clustering [22].

2.1 Dimensionality Reduction Based Clustering

Dimensionality reduction is widely used for clustering with high dimensional data. One common strategy is to do unsupervised dimensionality reduction before clustering, e.g. principal component analysis (PCA). However, PCA cannot lead to the separability of the data for clustering because it is unsupervised. Furthermore, since the dimensionality reduction is independent with the subsequent clustering, it does not necessarily achieve improvement in clustering performance. An alternative strategy is to integrate supervised dimensionality reduction with clustering in a joint framework, and do them simultaneously. In detail, clustering generates the class label for supervised dimensionality reduction, while supervised dimensionality reduction finds the discriminative subspace for clustering. The pioneer works are adaptive dimension reduction (ADR) [7] and adaptive subspace iteration (ASI) [15]. In [8], the authors proposed to combine linear discriminant analysis (LDA) and K-means into a coherent framework to adaptively select the most discriminative subspace, which unifies ADR and ASI as its special cases. Similar methods include discriminative cluster analysis (DCA)[14] and adaptive metric learning (AML) [24], which are all optimizing the following objective function

$$\max_{\mathbf{L},\mathbf{A}} f(\mathbf{L},\mathbf{A}) = \mathrm{tr}((\mathbf{A}(\mathbf{X}\mathbf{X}^T+\lambda\mathbf{I})\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}\mathbf{L}\mathbf{L}^T\mathbf{X}^T\mathbf{A}^T) \tag{1}$$

where $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]\in\mathbb{R}^{d\times n}$ is data matrix, $\mathbf{A}\in\mathbb{R}^{m\times d}$ is linear transformation which projects the d-dimensional input data to m-dimensional subspace, and $\mathbf{L}=\mathbf{F}(\mathbf{F}^T\mathbf{F})^{-1}\in\mathbb{R}^{n\times c}$ is weighted cluster assignment matrix with $\mathbf{F}_{ip}=1$ if $\mathbf{x}_i$ belongs to the p-th cluster and $\mathbf{F}_{ip}=0$ otherwise. Recently, the authors in [25] gave an insightful analysis on the discriminative clustering methods [14] [8] [24] in Eq.(1), showing that the optimal $\mathbf{L}^*$ to Eq.(1) solves the following optimization problem

$$\max_{\mathbf{L}} f(\mathbf{L}) = \mathrm{tr}(\mathbf{L}^T(\mathbf{I}-(\mathbf{I}+\frac{1}{\lambda}\mathbf{G})^{-1})\mathbf{L}) \tag{2}$$

where $\mathbf{G}=\mathbf{X}^T\mathbf{X}$ is Gram matrix and $\mathbf{I}\in\mathbb{R}^{n\times n}$ is the identity matrix. Eq.(2) is referred to as discriminative K-means (DisKM). The most appealing property of DisKM is that the linear transformation $\mathbf{A}$ is omitted. In other words, the problem in Eq.(2) does not involve explicit dimensionality reduction. And it can be seen as kernel K-means with a specific kernel matrix $\mathbf{I}-(\mathbf{I}+\frac{1}{\lambda}\mathbf{G})^{-1}$.

All the methods introduced above are all along the line of K-means, while there are many state of the art clustering methods, which outperform K-means significantly, e.g. spectral clustering [19] [17] and maximum margin clustering (MMC) [22] [23]. In the following, we will briefly introduce MMC.

2.2 Maximum Margin Clustering

Maximum margin clustering (MMC) [22] [23] is a most recently proposed clustering method which extends maximum margin principle in support vector machine (SVM) [18] to unsupervised learning setting. Since the class labels are unknown in unsupervised learning, MMC tries to find a cluster assignment of the data points, together with a hyperplane classifier, such that the resultant margin is maximized among all possible cluster assignments.

Let us recall support vector machine (SVM) [18] first. Given a data set $\mathcal{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}\in\mathbb{R}^d$, and their class labels $\mathbf{y}=[y_1,\ldots,y_n]^T\in\{-1,+1\}^n$, SVM finds a hyperplane $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b$ by solving the following optimization problem

$$\begin{aligned}\min_{\mathbf{w},b,\{\xi_i\}}\quad & \frac{1}{2}||\mathbf{w}||^2+\frac{C}{n}\sum_{i=1}^n\xi_i\\ \text{s.t.}\quad & y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i,\xi_i\geq 0,\\ & i=1,\ldots,n\end{aligned} \tag{3}$$

Different from SVM, where the class labels are given, binary class MMC [22] aims to find the best class assignment $\mathbf{y}=[y_1,\ldots,y_n]^T\in\{\pm 1\}^n$ such that a SVM trained on $\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}$ yields the largest margin. It can be formulated mathematically as the following optimization problem

$$\begin{aligned}\min_{\mathbf{w},b,\{\xi_i\},\mathbf{y}}\quad & \frac{1}{2}||\mathbf{w}||^2+\frac{C}{n}\sum_{i=1}^n\xi_i\\ \text{s.t.}\quad & y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq 1-\xi_i,\xi_i\geq 0,\\ & -l\leq\sum_{i=1}^n(\mathbf{w}^T\mathbf{x}_i+b)\leq l\\ & i=1,\ldots,n\end{aligned} \tag{4}$$

where $\sum_{i=1}^n\xi_i$ is divided by n to better capture how C scales with the data set size. Note that the second constraint is often known as the class balance constraint. It is introduced to avoid the trivially optimal solution which assigns all data points to the same class and thus achieves infinite margin. l is a constant controlling the class imbalance.

According to [23] [6], binary class MMC can be extended to multi-class setting. Multi-class MMC aims to find the best class assignment $\mathbf{y}=[y_1,\ldots,y_n]^T\in\{1,\ldots,c\}^n$ such that a SVM trained on $\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}$ yields the largest margin, i.e.

$$\begin{aligned}\min_{\{\mathbf{w}_p\},\{\xi_i\},\mathbf{y}}\quad & \frac{1}{2}\sum_{p=1}^c||\mathbf{w}_p||_2^2+\frac{C}{n}\sum_{i=1}^n\xi_i\\ \text{s.t.}\quad & \mathbf{w}_{y_i}^T\mathbf{x}_i-\mathbf{w}_r^T\mathbf{x}_i+\delta_{y_i,r}\geq 1-\xi_i,\xi_i\geq 0,\\ & -l\leq\sum_{i=1}^n\mathbf{w}_p^T\mathbf{x}_i-\sum_{i=1}^n\mathbf{w}_q^T\mathbf{x}_i\leq l\\ & i=1,\ldots,n,p,q,r=1,\ldots,c,\end{aligned} \tag{5}$$

where $\delta_{p,q}=1$ if $p=q$ and 0 otherwise. Note that the second constraint is the class balance constraint.

In the past years, many algorithms have been proposed for solving MMC, e.g. semi-definite programming (SDP)

based [22] [23] [20], support vector regression (SVR) based [26], cutting plane based [27] [28], multiple kernel learning based [16] and stochastic search based [10]. On the other hand, some variants of MMC are also proposed most recently, e.g. MMC with pairwise constraints [11]. However, all these methods perform MMC in the input space. When the dimensionality of the data is very high, the *curse of the dimensionality* mentioned above may occur. Hence in our study, we aim to combine dimensionality reduction with MMC in a unified framework.

3. BINARY CLASS SUBSPACE MAXIMUM MARGIN CLUSTERING

In this section, we will present binary class subspace maximum margin clustering (SMMC).

3.1 Objective

When the dimensionality of the data is very high, we aim to learn a subspace, in which maximum margin clustering is performed. To achieve this, we introduce a linear transformation $\mathbf{A} \in \mathbb{R}^{m\times d}$, projecting the data points from the input space to a subspace. Rather than doing dimensionality reduction and clustering successively, we turn to do dimensionality reduction and clustering simultaneously in a joint framework. The objective of binary class SMMC is to learn a linear transformation $\mathbf{A}$, find a cluster assignment $\mathbf{y} \in \{\pm 1\}^n$ of the data points in the subspace spanned by $\mathbf{A}$, together with the hyperplane classifier $\mathbf{w}$, such that the resultant margin is maximized among all possible cluster assignments and all possible subspaces. It is mathematically formulated as the following optimization problem

$$\begin{aligned} \min_{\mathbf{w},b,\{\xi_i\},\mathbf{y},\mathbf{A}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{A}\mathbf{x}_i + b) \geq 1-\xi_i, \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{A}\mathbf{x}_i + b) \leq l \\ & \mathbf{A}\mathbf{A}^T = \mathbf{I} \\ & i = 1,\ldots,n \end{aligned} \tag{6}$$

Note that $||\mathbf{w}||_2^2 = ||\mathbf{w}||_F^2$ where $||\cdot||_F$ is Frobenius norm. For $p, q \geq 1$, (p,q)-norm of matrix $\mathbf{W}$ is defined as $||\mathbf{W}||_{p,q} = (\sum_i^d ||\mathbf{w}^i||_p^q)^{\frac{1}{q}}$, where $\mathbf{w}^i$ is the i-th row of $\mathbf{W}$. It is easy to show that Frobenius norm is $(2,2)$-norm.

In Eq.(6), $\mathbf{w}$ and $\mathbf{A}$ are coupled together in the constraints. Moreover, the orthonormal constraint $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ is non-convex. To tackle the above problem, we replace the L_2 norm on $\mathbf{w}$ with $(2,1)$-norm, and we have the following theorem.

Theorem 3.1 *Eq.(6) using $(2,1)$-norm for $\mathbf{w}$ is equivalent to the following problem*

$$\begin{aligned} \min_{\mathbf{u},b,\{\xi_i\},\mathbf{y},\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\mathbf{u}^T\mathbf{D}^+\mathbf{u} + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i(\mathbf{u}^T\mathbf{x}_i + b) \geq 1-\xi_i, \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i + b) \leq l \\ & i = 1,\ldots,n \end{aligned} \tag{7}$$

where $\mathbf{u} = \mathbf{A}^T\mathbf{w}$, $\mathbf{D}^+$ is the pseudo-inverse of $\mathbf{D}$, $\mathcal{D} = \{\mathbf{D}|tr(\mathbf{D}) \leq 1, range(\mathbf{U}) \subseteq range(\mathbf{D}), \mathbf{D} \in \mathbb{S}_+^d\}$, $range(\mathbf{D})$ denotes the set $\{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} = \mathbf{D}\mathbf{z}, for\ some \mathbf{z} \in \mathbb{R}^d\}$, $\mathbb{S}_+^d$ denotes the set of positive semi-definite real symmetric matrices.

PROOF. Please refer to the proof of Theorem 4.1. □

It is worth noting that when $\mathbf{D}^+ = \mathbf{I}$, Eq.(7) degenerates to standard binary-class maximum margin clustering in Eq.(4). The most appealing property of the problem in Eq.(7) is that the linear transformation $\mathbf{A}$ is omitted, hence there is no need for tuning the dimensionality of the subspace. And learning the subspace is transformed to learning the positive semi-definite matrix $\mathbf{D}$. To some extent, it is similar with DisKM in Eq.(2) which omits the linear transformation $\mathbf{A}$ in Eq.(1).

3.2 Optimization

In the following, we will present an algorithm for solving the problem in Eq.(7).

Eq.(7) is a mixed integer programming, which is difficult to solve. Fortunately, we have the following theorem.

Theorem 3.2 *Eq.(7) is equivalent to Eq.(8)*

$$\begin{aligned} \min_{\mathbf{u},b,\{\xi_i\},\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\mathbf{u}^T\mathbf{D}^+\mathbf{u} + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ s.t. \quad & |\mathbf{u}^T\mathbf{x}_i + b| \geq 1-\xi_i, \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i + b) \leq l \\ & i = 1,\ldots,n \end{aligned} \tag{8}$$

where $y_i = sign(\mathbf{u}^T\mathbf{x}_i + b)$.

PROOF. Please refer to [27]. □

One challenge in solving Eq.(8) is that the number of variables is very large. To tackle this, we adopt the strategy in [12] to transform the n-slack formulation in Eq.(8) to 1-slack formulation.

Theorem 3.3 *Eq.(8) is equivalent to Eq.(9), with $\xi^* = \frac{1}{n}\sum_{i=1}^{n}\xi_i^*$*

$$\begin{aligned} \min_{\mathbf{u},b,\xi,\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\mathbf{u}^T\mathbf{D}^+\mathbf{u} + C\xi \\ s.t. \quad & \forall \mathbf{c} \in \{0,1\}^n \\ & \frac{1}{n}\sum_{i=1}^{n}c_i|\mathbf{u}^T\mathbf{x}_i + b| \geq \frac{1}{n}\sum_{i=1}^{n}c_i - \xi, \xi \geq 0 \\ & -l \leq \sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i + b) \leq l \\ & i = 1,\ldots,n \end{aligned} \tag{9}$$

where $\mathbf{c} = [c_1,\ldots,c_n]^T$.

PROOF. Please refer to [27]. □

Although the number of slack variables in Eq.(9) is greatly reduced by $n-1$, the number of constraints increases from n to 2^n. Fortunately, cutting plane technique [13] enables us to deal with this problem, which keeps a polynomial sized

subset Ω of working constraints and computes the optimal solution to Eq.(9) subject to the constraints in Ω. In detail, the algorithm adds the most violated constraint in Eq.(9) into Ω in each iteration. In this way, a successively strengthening approximation of the original problem is solved. And the algorithm terminates when no constraints in Eq.(9) is violated by more than ϵ. The remaining thing is how to find the most violated constraint in each iteration. Since the feasibility of a constraint is measured by the corresponding value of ξ, the most violated constraint is the one which owns the largest ξ. The following theorem gives the calculation of the most violated constraint.

Theorem 3.4 *The most violated constraint could be calculated as follows*

$$c_i = \begin{cases} 1, & if |\mathbf{u}^T\mathbf{x}_i + b| < 1, \\ 0, & otherwise. \end{cases} \tag{10}$$

PROOF. Please refer to [27]. □

In each iteration of the cutting plane algorithm, we need to solve Eq.(9) to obtain the optimal hyperplanes classifier under the current working constraint set Ω. However, the first constraint in Eq.(9) is non-convex. Note that the non-convex constraint can be written as the difference of two convex functions. As a result, we can use constrained concave-convex procedure (CCCP) [21] to solve this kind of problem, which owns well-studied convergence properties. Furthermore, although $\frac{1}{n}\sum_{i=1}^{n} c_i|\mathbf{u}^T\mathbf{x}_i + b|$ is convex, it is a non-smooth function of $(\mathbf{u}, b)$. In order to use CCCP, we need to use the sub-gradients [3] instead of gradients as follows,

$$\begin{aligned} & \partial_{\mathbf{u}}\left(\frac{1}{n}\sum_{i=1}^{n} c_i|\mathbf{u}^T\mathbf{x}_i + b|\right)\Big|_{\mathbf{u}=\mathbf{u}^{(t)}} \\ = & \frac{1}{n}\sum_{i=1}^{n} c_i \text{sign}(\mathbf{u}^{(t)T}\mathbf{x}_i + b^{(t)})\mathbf{x}_i \\ & \partial_b\left(\frac{1}{n}\sum_{i=1}^{n} c_i|(\mathbf{u})^T\mathbf{x}_i + b|\right)\Big|_{b=b^{(t)}} \\ = & \frac{1}{n}\sum_{i=1}^{n} c_i \text{sign}(\mathbf{u}^{(t)T}\mathbf{x}_i + b^{(t)}) \end{aligned} \tag{11}$$

Given an initial point $(\mathbf{u}^{(0)}, b^{(0)})$, CCCP computes $(\mathbf{u}^{(t+1)}, b^{(t+1)})$ from $(\mathbf{u}^{(t)}, b^{(t)})$ by replacing $\frac{1}{n}\sum_{i=1}^{n} c_i|\mathbf{u}^T\mathbf{x}_i + b|$ in the constraint with its first order Taylor expansion at $(\mathbf{u}^{(t)}, b^{(t)})$. Thus we obtain the following quadratic programming (QP) problem

$$\begin{aligned} \min_{\{\mathbf{u},b,\xi,\mathbf{D}\in\mathcal{D}\}} \quad & \frac{1}{2}\mathbf{u}^T\mathbf{D}^+\mathbf{u} + C\xi \\ \text{s.t.} \quad & \forall \mathbf{c} \in \Omega \\ & \frac{1}{n}\sum_{i=1}^{n} c_i \text{sign}(\mathbf{u}^{(t)T}\mathbf{x}_i + b^{(t)})(\mathbf{u}^T\mathbf{x}_i + b) \\ & \geq \frac{1}{n}\sum_{i=1}^{n} c_i - \xi, \xi \geq 0, \\ & -l \leq \sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i + b) \leq l \\ & i = 1, \ldots, n \end{aligned} \tag{12}$$

The following theorem characterizes the joint convexity of the problem in Eq.(12).

Theorem 3.5 *The optimization problem in Eq.(12) is jointly convex on* $\mathbf{U}$ *and* $\mathbf{D}$.

PROOF. Please refer to the proof of Theorem 4.5. □

According to this theorem, we can use alternating minimization to obtain the global optimum of Eq.(12). In other words, we fix one of the two optimization variables ($\mathbf{u}$ and $\mathbf{D}$), and solve the other one in terms of the fixed one.

In detail, fixing $\mathbf{D}^{(t)}$, $\mathbf{u}^{(t+1)}$ can be solved via QP as follows

$$\begin{aligned} \min_{\{\mathbf{u},b,\xi\}} \quad & \frac{1}{2}\mathbf{u}^T\mathbf{D}^{(t)+}\mathbf{u} + C\xi \\ \text{s.t.} \quad & \forall \mathbf{c} \in \Omega \\ & \frac{1}{n}\sum_{i=1}^{n} c_i \text{sign}(\mathbf{u}^{(t)T}\mathbf{x}_i + b^{(t)})(\mathbf{u}^T\mathbf{x}_i + b) \\ & \geq \frac{1}{n}\sum_{i=1}^{n} c_i - \xi, \xi \geq 0, \\ & -l \leq \sum_{i=1}^{n}(\mathbf{u}^T\mathbf{x}_i + b) \leq l \\ & i = 1, \ldots, n \end{aligned} \tag{13}$$

Conversely, fixing $\mathbf{u}^{(t+1)}$, $\mathbf{D}^{(t+1)}$ can be solved by the following problem,

$$\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\mathbf{u}^{(t+1)T}\mathbf{D}^+\mathbf{u}^{(t+1)} \tag{14}$$

The following theorem characterizes the optimal solution to Eq.(14)

Theorem 3.6 *Let* $\mathbf{C} = \mathbf{u}\mathbf{u}^T$, *the optimal solution of Eq.(14) is*

$$\mathbf{D} = \frac{\mathbf{C}^{\frac{1}{2}}}{tr(\mathbf{C}^{\frac{1}{2}})} \tag{15}$$

and the optimal value equals $(tr(\mathbf{C}^{\frac{1}{2}}))^2$

PROOF. Please refer to the proof of Theorem 4.6. □

As for the termination criterion of CCCP (alternating minimization), we check if the difference in objective values from two successive iterations is less than $\alpha\%$. In our experiments, we set $\alpha = 1$.

In summary, we present the whole algorithm of optimizing Eq.(7) in Algorithm 1.

3.3 Theoretical Analysis

The convergence of Algorithm 1 is theoretically guaranteed by the following theorem.

Theorem 3.7 *For any* $\epsilon > 0$, $C > 0$, *and any data set* $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, *the Algorithm 1 terminates after adding at most* $\frac{CR}{\epsilon^2}$ *constraints, where* R *is a constant number independent of* n *and* d.

PROOF. Please refer to [27]. □

And the computational complexity of Algorithm 1 is characterized in the following theorem.

Algorithm 1 Binary Class Subspace Maximum Margin Clustering

Input: C, l, ϵ and α, $\Omega = \emptyset$;
Output: $\{\mathbf{u}_p\}$;
Set $\mathbf{D} = \frac{1}{d}\mathbf{I}$, initialize $(\mathbf{u}^{(0)}, b^{(0)})$ using K-means;
repeat
  repeat
    Solve for $(\mathbf{u}^{(t+1)}, b^{(t+1)}, \xi^{(t+1)})$ as the solution to Eq.(13) under the working constraint set Ω;
    Solve for $\mathbf{D}^{(t+1)}$ as in Eq.(15);
  until the stopping criterion is satisfied
  Select the most violated constraint $\mathbf{c}$ as in Eq.(10), and set $\Omega = \Omega \cup \mathbf{c}$;
until the newly selected constraint $\mathbf{c}$ is violated by no more than ϵ

Theorem 3.8 *The computational complexity of algorithm 1 is* $O(\frac{CRd^2}{\epsilon^2})$

PROOF. In each iteration of CCCP, the computational complexity of solving the QP in Eq.(13) is $O(|\Omega|^2 dn)$ where $|\Omega|$ is the size of the working set. And the computational complexity of Eq.(15) is $O(d^2)$. Hence the computational complexity is about $O(d^2)$ in each iteration. Moreover, we observed that CCCP takes less than 10 iterations in each round of cutting plane, then according to Theorem 3.7, the overall computational complexity of Algorithm 1 is $O(\frac{CRd^2}{\epsilon^2})$. This completes the proof. □

4. MULTI-CLASS SUBSPACE MAXIMUM MARGIN CLUSTERING

In this section, we will extend binary class subspace maximum margin clustering to multi-class subspace maximum margin clustering.

4.1 Objective

When it comes to multi-class clustering setting, the objective of SMMC is to learn a linear transformation $\mathbf{A}$, find a cluster assignment $\mathbf{y} \in \{1, \ldots, c\}^n$ of the data points in the subspace spanned by $\mathbf{A}$, together with the hyperplane classifier $\mathbf{w}_p, p = 1, \ldots, c$, such that the resultant margin is maximized among all possible cluster assignments and all possible subspaces, i.e.

$$\begin{aligned} \min_{\{\mathbf{w}_p\},\{\xi_i\},\mathbf{y}} \quad & \frac{1}{2}\sum_{p=1}^{c}||\mathbf{w}_p||_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \mathbf{w}_{y_i}^T\mathbf{A}\mathbf{x}_i - \mathbf{w}_r^T\mathbf{A}\mathbf{x}_i + \delta_{y_i,r} \geq 1 - \xi_i, \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n}\mathbf{w}_p^T\mathbf{A}\mathbf{x}_i - \sum_{i=1}^{n}\mathbf{w}_q^T\mathbf{A}\mathbf{x}_i \leq l \\ & \mathbf{A}\mathbf{A}^T = \mathbf{I} \\ & i = 1, \ldots, n, p, q, r = 1, \ldots, c \end{aligned} \tag{16}$$

Note that $\sum_{p=1}^{c}||\mathbf{w}_p||_2^2 = ||\mathbf{W}||_F^2$ where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_c]$ and $||\cdot||_F$ is Frobenius norm. For $p, q \geq 1$, (p, q)-norm of matrix $\mathbf{W}$ is defined as $||\mathbf{W}||_{p,q} = (\sum_i^d ||\mathbf{w}^i||_p^q)^{\frac{1}{q}}$, where $\mathbf{w}^i$ is the i-th row of $\mathbf{W}$. It is easy to show that Frobenius norm is $(2, 2)$-norm.

In Eq.(16), $\mathbf{w}_p$ and $\mathbf{A}$ are coupled together in the constraints. Moreover, the orthonormal constraint $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ is non-convex. Again, we replace the Frobenious norm on $\mathbf{W}$ with $(2, 1)$-norm, and we have the following theorem.

Theorem 4.1 *Eq.(16) using* $(2, 1)$*-norm for* $\mathbf{W}$ *is equivalent to the following problem*

$$\begin{aligned} \min_{\{\mathbf{u}_p\},\{\xi_i\},\mathbf{y},\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p + \frac{C}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \mathbf{u}_{y_i}^T\mathbf{x}_i - \mathbf{u}_r^T\mathbf{x}_i + \delta_{y_i,r} \geq 1 - \xi_i, \xi_i \geq 0, \\ & -l \leq \sum_{i=1}^{n}\mathbf{u}_p^T\mathbf{x}_i - \sum_{i=1}^{n}\mathbf{u}_q^T\mathbf{x}_i \leq l \\ & i = 1, \ldots, n, p, q, r = 1, \ldots, c \end{aligned} \tag{17}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_c] = \mathbf{A}^T[\mathbf{w}_1, \ldots, \mathbf{w}_c] = \mathbf{A}^T\mathbf{W}$, $\mathbf{D}^+$ *is the pseudo-inverse of* $\mathbf{D}$, $\mathcal{D} = \{\mathbf{D}|tr(\mathbf{D}) \leq 1, range(\mathbf{U}) \subseteq range(\mathbf{D}), \mathbf{D} \in \mathbb{S}_+^d\}$, $range(\mathbf{D})$ *denotes the set* $\{\mathbf{x} \in \mathbb{R}^d | \mathbf{x} = \mathbf{D}\mathbf{z}, for\ some \mathbf{z} \in \mathbb{R}^d\}$, $\mathbb{S}_+^d$ *denotes the set of positive semi-definite real symmetric matrices.*

PROOF. The proof is based on [1]. let $\mathbf{D} = \mathbf{A}^T\text{diag}(\frac{||\mathbf{w}^i||_2}{||\mathbf{W}||_{2,1}})\mathbf{A}$, then

$$\begin{aligned} \sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p &= \text{tr}(\mathbf{U}^T\mathbf{D}^+\mathbf{U}) \\ &= \text{tr}(\mathbf{W}^T\mathbf{A}\mathbf{A}^T\text{diag}(\frac{||\mathbf{w}^i||_2}{||\mathbf{W}||_{2,1}})^+\mathbf{A}\mathbf{A}^T\mathbf{W}) \\ &= \text{tr}(\text{diag}(\frac{||\mathbf{W}||_{2,1}}{||\mathbf{w}^i||_2})\mathbf{W}\mathbf{W}^T) \\ &= ||\mathbf{W}||_{2,1}^2 \end{aligned}$$

Hence the optimal value of problem in Eq.(17) is less than or equal to the optimal value of problem in Eq.(16) using $(2, 1)$-norm.

Conversely, let $\mathbf{D} = \mathbf{A}^T\text{diag}(\boldsymbol{\lambda})\mathbf{A}$ where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_d]^T \in \mathbb{R}^d$, then

$$\begin{aligned} \sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p &= \text{tr}(\mathbf{U}^T\mathbf{D}^+\mathbf{U}) \\ &= \text{tr}(\mathbf{W}^T\mathbf{A}\mathbf{A}^T\text{diag}(\lambda)^+\mathbf{A}\mathbf{A}^T\mathbf{W}) \\ &= \text{tr}(\text{diag}(\lambda)^+\mathbf{W}\mathbf{W}^T) \\ &= \sum_{p=1}^{c}\lambda_i^{-1}||\mathbf{w}^i||_2^2 \\ &\geq \sum_{p=1}^{c}\lambda_i^{-1}||\mathbf{w}^i||_2^2\sum_{p=1}^{c}\lambda_i \\ &\geq (\sum_{p=1}^{c}\lambda_i^{-\frac{1}{2}}||\mathbf{w}^i||_2\lambda_i^{\frac{1}{2}})^2 \\ &= ||\mathbf{W}||_{2,1}^2 \end{aligned}$$

The last inequality holds based on the Cauchy-Schwarz Inequality. Hence the optimal value of problem in Eq.(16) using $(2, 1)$-norm is less than or equal to the optimal value of problem in Eq.(17).

In summary, the optimal value of problem in Eq.(16) equals to the optimal value of problem in Eq.(17). This completes the proof. □

It is worth noting that when $\mathbf{D}^+ = \mathbf{I}$, Eq.(17) degenerates to standard multi-class maximum margin clustering in Eq.(5).

The most appealing property of the problem in Eq.(17) is that the linear transformation $\mathbf{A}$ is omitted, hence there is no need for tuning the dimensionality of the subspace. And learning the subspace is transformed to learning the positive semi-definite matrix $\mathbf{D}$.

4.2 Optimization

In the following, we will present an algorithm for solving the problem in Eq.(17). Again, Eq.(17) is a mixed integer programming, which is difficult to solve. Fortunately, we have the following theorem.

Theorem 4.2 *Eq.(17) is equivalent to Eq.(18)*

$$\begin{aligned}\min_{\{\mathbf{u}_p\},\{\xi_i\},\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p+\frac{C}{n}\sum_{i=1}^{n}\xi_i \\ s.t. \quad & \sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}+z_{ir}-\mathbf{u}_r^T\mathbf{x}_i\geq 1-\xi_i, \\ & -l\leq\sum_{i=1}^{n}\mathbf{u}_p^T\mathbf{x}_i-\sum_{i=1}^{n}\mathbf{u}_q^T\mathbf{x}_i\leq l \\ & i=1,\ldots,n,p,q,r=1,\ldots,c \end{aligned} \tag{18}$$

where $I(\cdot)$ is the indicator function, $z_{ip}=\prod_{q=1,q\neq p}^{c} I_{(\mathbf{u}_p^T\mathbf{x}_i>\mathbf{u}_q^T\mathbf{x}_i))}$, and the label for data point $\mathbf{x}_i$ is determined as $y_i=\arg\max_p \mathbf{u}_p^T\mathbf{x}_i=\sum_{p=1}^{c}pz_{ip}$.

Proof. Please refer to [28]. □

Again, we adopt the strategy in [12] to transform the n-slack formulation in Eq.(18) to 1-slack formulation.

Theorem 4.3 *Eq.(18) is equivalent to Eq.(19), with $\xi^*=\frac{1}{n}\sum_{i=1}^{n}\xi_i^*$*

$$\begin{aligned}\min_{\{\mathbf{u}_p\},\xi,\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\sum_{p=1}^{c}(\mathbf{u}_p)^T\mathbf{D}^+\mathbf{u}_p+C\xi \\ s.t. \quad & \mathbf{c}_i\in\{\mathbf{e}_0,\mathbf{e}_1,\ldots,\mathbf{e}_c\} \\ & \frac{1}{n}\sum_{i=1}^{n}[\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}+\sum_{p=1}^{c}c_{ip}(z_{ip}-\mathbf{u}_p^T\mathbf{x}_i)] \\ & \geq\frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e}-\xi, \\ & -l\leq\sum_{i=1}^{n}\mathbf{u}_p^T\mathbf{x}_i-\sum_{i=1}^{n}\mathbf{u}_q^T\mathbf{x}_i\leq l \\ & i=1,\ldots,n,p,q=1,\ldots,c \end{aligned} \tag{19}$$

where $\mathbf{e}_p$ is a $c\times 1$ vector with only the p-th element being 1 and others 0, $\mathbf{e}_0$ is the $c\times 1$ zero vector and $\mathbf{e}$ is the all one vector.

Proof. Please refer to [28]. □

Although the number of variables in Eq.(19) is greatly reduced by $n-1$, the number of constraints increases from nc to $(c+1)^n$. Again, we adopt cutting plane technique [13], which keeps a polynomial sized subset Ω of working constraints and computes the optimal solution to Eq.(19) subject to the constraints in Ω. More concretely, the algorithm adds the most violated constraint in Eq.(19) into Ω in each iteration. In this way, a successively strengthening approximation of the original problem is constructed and solved. The algorithm terminates when no constraints in Eq.(19) is violated by more than ϵ. The key point is how to find the most violated constraint in each iteration. Since the feasibility of a constraint is measured by the corresponding value of ξ, the most violated constraint is the one which would lead to the largest ξ. The following theorem gives the computation of the most violated constraint.

Theorem 4.4 *Define $p^*=\arg\max_p(\mathbf{u}_p^T\mathbf{x}_i)$ and $r^*=\arg\max_{r\neq p^*}(\mathbf{u}_r^T\mathbf{x}_i)$ for $i=1,2,\ldots,n$, the most violated constraint could be calculated as follows*

$$\mathbf{c}_i=\begin{cases}\mathbf{e}_{r^*}, & if(\mathbf{u}_{p^*}^T\mathbf{x}_i-\mathbf{u}_{r^*}^T\mathbf{x}_i)<1,\\ 0, & otherwise.\end{cases} \tag{20}$$

Proof. Please refer to [28]. □

In each iteration of the cutting plane algorithm, we need to solve Eq.(19) to obtain the optimal hyperplane classifier under the current working constraint set Ω. However, the first constraint in Eq.(19) is non-convex. Again, the non-convex constraint can be written as the difference of two convex functions. And we can use constrained concave-convex procedure (CCCP) [21] to solve this kind of problem. Furthermore, although $\frac{1}{n}\sum_{i=1}^{n}\{\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}+\sum_{p=1}^{c}c_{ip}z_{ip}\}$ is convex, it is a non-smooth function of $(\mathbf{u}_1,\ldots,\mathbf{u}_c)$. To use CCCP, we need to compute the sub-gradients [3] as follows,

$$\begin{aligned}&\partial_{\mathbf{u}_r}\{\frac{1}{n}\sum_{i=1}^{n}[\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}+\sum_{p=1}^{c}c_{ip}z_{ip}]\}|_{\mathbf{u}_r=\mathbf{u}_r^{(t)}} \\ &=\frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e}z_{ip}^{(t)}\mathbf{x}_i\end{aligned} \tag{21}$$

Given an initial point $(\mathbf{u}_1^{(0)},\ldots,\mathbf{u}_c^{(0)})$, CCCP computes $(\mathbf{u}_1^{(t+1)},\ldots,\mathbf{u}_c^{(t+1)})$ from $(\mathbf{u}_1^{(t)},\ldots,\mathbf{u}_c^{(t)})$ by replacing $\frac{1}{n}\sum_{i=1}^{n}\{\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}+\sum_{p=1}^{c}c_{ip}z_{ip}\}$ in the constraint with its first order Taylor expansion at $(\mathbf{u}_1^{(t)},\ldots,\mathbf{u}_c^{(t)})$. Thus we obtain the following quadratic programming (QP) problem

$$\begin{aligned}\min_{\{\mathbf{u}_p\},\xi,\mathbf{D}\in\mathcal{D}} \quad & \frac{1}{2}\sum_{p=1}^{c}(\mathbf{u}_p)^T\mathbf{D}^+\mathbf{u}_p+C\xi \\ \text{s.t.} \quad & \forall[\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_n]\in\Omega \\ & \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e}-\xi+\frac{1}{n}\sum_{i=1}^{n}\sum_{p=1}^{c}c_{ip}\mathbf{u}_p^T\mathbf{x}_i \\ & -\frac{1}{n}\sum_{i=1}^{n}[\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}^{(t)}+\sum_{p=1}^{c}c_{ip}z_{ip}^{(t)}]\leq 0 \\ & -l\leq\sum_{i=1}^{n}\mathbf{u}_p^T\mathbf{x}_i-\sum_{i=1}^{n}\mathbf{u}_q^T\mathbf{x}_i\leq l \\ & i=1,\ldots,n,p,q=1,\ldots,c \end{aligned} \tag{22}$$

The following theorem characterizes the joint convexity of the problem in Eq.(22).

Theorem 4.5 *The optimization problem in Eq.(22) is jointly convex on $\mathbf{U}$ and $\mathbf{D}$.*

Proof. The proof is based on [5]. It is trivial to show that $\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p$ is jointly convex on $\mathbf{U}$ and $\mathbf{D}$ if and

only if $\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p, 1 \leq p \leq c$ is joint convex on $\mathbf{U}$ and $\mathbf{D}$. According to [3], a function f is convex if and only if the epigraph of the function, denoted as epi$f$ is a convex set, where

$$\text{epi}f = \{(\mathbf{X}, t) \in \mathcal{X} \times \mathbb{R} | \mathbf{X} \in \mathcal{X}, f(\mathbf{X}) \preceq t\}$$

Here $f(\mathbf{u}_p, \mathbf{D}) = \mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p$, so the epigraph of f is

$$\begin{aligned} & \text{epi}f(\mathbf{u}_p, \mathbf{D}) \\ = \; & \{(\mathbf{u}_p, \mathbf{D}, t) | \text{range}(\mathbf{U}) \subseteq \text{range}(\mathbf{D}), \mathbf{D} \in \mathbb{S}_d^+, \mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p \preceq t\} \end{aligned}$$

According to Schur Complement Theory [3],

$$\left[\begin{array}{cc} \mathbf{D} & \mathbf{u}_p \\ \mathbf{u}_p^T & t \end{array} \right] \succeq 0 \Leftrightarrow \left\{ \begin{array}{c} \mathbf{D} \succeq 0, \\ t - \mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p \succeq 0, \\ (\mathbf{I} - \mathbf{D}\mathbf{D}^+)\mathbf{u}_p = 0 \end{array} \right.$$

Since the left-hand-side of the above equality is a positive semi-definite cone, it is convex. Hence the right-hand-side of the equality is convex. Note that the right-hand-side of the equality corresponds to the epigraph of function $f(\mathbf{u}_p, \mathbf{D})$, this completes the proof. □

According to this theorem, we can use alternating minimization to obtain the global optimum of Eq.(22). In other words, we fix one of the two optimization variables ($\mathbf{U}$ and $\mathbf{D}$), and solve the other one in terms of the fixed one.

In detail, fixing $\mathbf{D}^{(t)}$, $\{\mathbf{u}_p^{(t+1)}\}$ can be solved via QP as follows,

$$\begin{aligned} \min_{\{\mathbf{u}_p\},\xi} \quad & \frac{1}{2}\sum_{p=1}^{c}(\mathbf{u}_p)^T\mathbf{D}^{(t)+}\mathbf{u}_p + C\xi \\ \text{s.t.} \quad & \forall[\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n] \in \Omega \\ & \frac{1}{n}\sum_{i=1}^{n}\mathbf{c}_i^T\mathbf{e} - \xi + \frac{1}{n}\sum_{i=1}^{n}\sum_{p=1}^{c} c_{ip}\mathbf{u}_p^T\mathbf{x}_i \\ & -\frac{1}{n}\sum_{i=1}^{n}[\mathbf{c}_i^T\mathbf{e}\sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{x}_i z_{ip}^{(t)} + \sum_{p=1}^{c} c_{ip}z_{ip}^{(t)}] \leq 0 \\ & -l \leq \sum_{i=1}^{n}\mathbf{u}_p^T\mathbf{x}_i - \sum_{i=1}^{n}\mathbf{u}_q^T\mathbf{x}_i \leq l \\ & i = 1, \ldots, n, p, q = 1, \ldots, c \end{aligned} \tag{23}$$

Conversely, fixing $\{\mathbf{u}_p^{(t+1)}\}$, $\mathbf{D}^{(t+1)}$ can be solved by the following problem,

$$\min_{\mathbf{D}\in\mathcal{D}} \frac{1}{2}\sum_{p=1}^{c}\mathbf{u}_p^{(t+1)T}\mathbf{D}^+\mathbf{u}_p^{(t+1)} \tag{24}$$

The following theorem characterizes the optimal solution to Eq.(24)

Theorem 4.6 *Let $\mathbf{C} = \mathbf{U}\mathbf{U}^T$, the optimal solution of Eq.(24) is*

$$\mathbf{D} = \frac{\mathbf{C}^{\frac{1}{2}}}{tr(\mathbf{C}^{\frac{1}{2}})} \tag{25}$$

and the optimal value equals $(tr(\mathbf{C}^{\frac{1}{2}}))^2$

Proof. The proof is based on [1]. Let $\mathbf{D} = \mathbf{A}\text{diag}(\boldsymbol{\lambda})\mathbf{A}^T$ where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_d] \in \mathbb{R}^d$, then

$$\begin{aligned} \sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p &= \text{tr}(\mathbf{U}^T\mathbf{D}^+\mathbf{U}) \\ &= \text{tr}(\mathbf{U}^T\mathbf{A}\text{diag}(\lambda)^+\mathbf{A}^T\mathbf{U}) \\ &= \text{tr}(\text{diag}(\lambda)^+\mathbf{A}^T\mathbf{U}\mathbf{U}^T\mathbf{A}) \\ &= \sum_{p=1}^{c}\frac{\mathbf{a}_i^T\mathbf{U}\mathbf{U}^T\mathbf{a}_i}{\lambda_i} \\ &\geq (\sum_{p=1}^{c}||\mathbf{U}^T\mathbf{a}_i||_2)^2 \end{aligned}$$

The last inequality holds based on Lemma A.1. Next, we have

$$\begin{aligned} ||\mathbf{U}^T\mathbf{a}_i||_2^2 &= \mathbf{a}_i^T\mathbf{U}\mathbf{U}^T\mathbf{a}_i \\ &= \mathbf{a}_i^T\mathbf{C}\mathbf{a}_i \\ &= \text{tr}(\mathbf{a}_i^T\mathbf{C}\mathbf{a}_i)\text{tr}(\mathbf{a}_i^T\mathbf{a}_i) \\ &= \text{tr}(\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}})\text{tr}(\mathbf{a}_i\mathbf{a}_i^T) \\ &\geq \text{tr}(\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i\mathbf{a}_i^T) \\ &= \text{tr}(\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i) \\ &= (\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i)^2 \end{aligned}$$

since $\text{tr}(\mathbf{A})\text{tr}(\mathbf{B}) \geq \text{tr}(\mathbf{A}\mathbf{B})$ if $\mathbf{A}$ and $\mathbf{B}$ are positive semi-definite. The equality holds if and only if $\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i\mathbf{a}_i^T = \mu\mathbf{a}_i\mathbf{a}_i^T$ which implies that $\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i = \mu\mathbf{a}_i$, that is, $\mathbf{a}_i$ is an eigenvector of $\mathbf{C}^{\frac{1}{2}}$. The optimal μ is $\text{tr}(C^{\frac{1}{2}})$. Hence we obtain

$$\begin{aligned} \sum_{p=1}^{c}\mathbf{u}_p^T\mathbf{D}^+\mathbf{u}_p &\geq (\sum_{p=1}^{c}\mathbf{a}_i^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_i)^2 \\ &= (\text{tr}(\mathbf{A}\mathbf{C}^{\frac{1}{2}}\mathbf{A}^T))^2 \\ &= (\text{tr}(\mathbf{C}^{\frac{1}{2}}))^2 \end{aligned}$$

Consequently, the optimal $\mathbf{D} = \frac{\mathbf{C}^{\frac{1}{2}}}{\text{tr}(\mathbf{C}^{\frac{1}{2}})}$. This completes the proof. □

As for the termination criterion of CCCP (alternating minimization), we check if the difference in objective values from two successive iterations is less than $\alpha\%$. In our experiments, we set $\alpha = 1$.

In summary, we present the whole algorithm of optimizing Eq.(17) in Algorithm 2.

4.3 Theoretical Analysis

The convergence of Algorithm 2 is theoretically guaranteed by the following theorem.

Theorem 4.7 *For any $\epsilon > 0$, $C > 0$, and any data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the Algorithm 2 terminates after adding at most $\frac{CR}{\epsilon^2}$ constraints, where R is a constant number independent of n and d.*

Proof. Please refer to [28]. □

And the computational complexity of Algorithm 2 is characterized in the following theorem.

Theorem 4.8 *The computational complexity of Algorithm 2 is $O(\frac{CRrd^2}{\epsilon^2})$, where r is the rank of $\mathbf{U}$.*

Algorithm 2 Multi-Class Subspace Maximum Margin Clustering

Input: C, l, ϵ and α, $\Omega = \emptyset$;
Output: $\{\mathbf{u}_p\}$;
Set $\mathbf{D} = \frac{1}{d}\mathbf{I}$, initialize $\{\mathbf{u}_p^{(0)}\}$, $\{z_{ip}\}$ using K-means;
repeat
  repeat
    Solve for $\{\mathbf{u}_p^{(t+1)}, \xi^{(t+1)}\}$ as the solution to Eq.(23) under the working constraint set Ω;
    Solve for $\mathbf{D}^{(t+1)}$ as in Eq.(25);
  until the stopping criterion is satisfied
  Select the most violated constraint $\mathbf{c}$ as in Eq.(20), and set $\Omega = \Omega \cup \mathbf{c}$;
until the newly selected constraint $\mathbf{c}$ is violated by no more than ϵ

PROOF. In each iteration of CCCP, the computational complexity of solving the QP in Eq.(23) is $O(|\Omega|^2 dn)$ where $|\Omega|$ is the size of the working set, and the computational complexity of Eq.(25) is $O(rd^2)$, where r is the rank of $\mathbf{U}$. Hence the computational complexity is about $O(rd^2)$ in each iteration. Moreover, we observed that CCCP takes less than 10 iterations in each round of cutting plane, then according to Theorem 4.7, the overall computational complexity of Algorithm 2 is $O(\frac{CRrd^2}{\epsilon^2})$. This completes the proof. □

5. EXPERIMENTS

In our experiments, we will evaluate the proposed clustering method on benchmark data sets. We compare it with state of the art clustering methods, e.g. normalized cut (NCut) [19] and maximum margin clustering (MMC) [28]. We also compare it with dimensionality reduction based methods, e.g. PCA+K-means (PCAKM), adaptive dimensionality reduction (ADR) [7], adaptive subspace iteration (ASI) [15] and discriminative K-means (DisKM) [25].

5.1 Evaluation Metrics

To evaluate the clustering results, we adopt the performance measures used in [4]. These performance measures are the standard measures widely used for clustering.

Clustering Accuracy: Clustering Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. Clustering Accuracy is defined as follows:

$$Acc = \frac{\sum_{i=1}^{n} \delta(map(r_i), l_i)}{n}, \tag{26}$$

where r_i denotes the cluster label of $\mathbf{x}_i$, and l_i denotes the true class label, n is the total number of documents, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data set.

Normalized Mutual Information: The second measure is the Normalized Mutual Information (NMI), which is used for determining the quality of clusters. Given a clustering result, the NMI is estimated by

$$NMI = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{i,j} \log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i \log \frac{n_i}{n})(\sum_{j=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n})}}, \tag{27}$$

where n_i denotes the number of data contained in the cluster $\mathcal{C}_i (1 \le i \le c)$, $\hat{n}_j$ is the number of data belonging to the $\mathcal{L}_j (1 \le j \le c)$, and $n_{i,j}$ denotes the number of data that are in the intersection between the cluster $\mathcal{C}_i$ and the class $\mathcal{L}_j$. The larger the NMI is, the better the clustering result will be.

5.2 Data Sets

In order to evaluate the clustering methods, we use a subset of UCI database and three text data sets. These data sets have also been used in [27] [28].

UCI: A subset of UCI [2] machine learning benchmark database is selected to evaluate the binary class clustering, e.g. ionosphere, digits and Letter. Following [27], for the digits data set, we focus on those pairs (3 vs 8,1 vs 7, 2 vs 7, 8 vs 9) that are difficult to differentiate. For the letter data sets, we use their first 2 classes.

Cora[1]**:** For Cora data set, we select a subset containing the research papers of subfield data structure(DS), hardware and architecture (HA), machine learning (ML), operating system (OS) and programming language (PL).

WebKB[2]**:** The WebKB data set contains web pages gathered from 4 university computer science departments. There are about 8280 documents and they are divided into 7 categories: student, faculty, staff, course, project, department and other, among which student, faculty, course and project are four most populous entity-representing categories. We select a subset of about 6000 web pages.

20Newsgroup[3]**:** The topic *rec* containing *autos, motorcycles, baseball* and *hockey* was selected from the version 20news-18828. To evaluate the binary class clustering, we focus on those pairs (autos vs motorcycles (Text1), and baseball vs hockey (Text2)) that are difficult to differentiate. To evaluate the multi-class clustering, all of them are used, referred to as News4.

Table.1 summarizes the characteristics of the data sets used in this experiment.

Table 1: Description of the data sets

Data set	#Sample(n)	#Feature(d)	#Class(c)
ionosphere	351	34	2
digits	1797	64	2
Letter AvB	1555	16	2
Text1	1980	8014	2
Text2	1990	8014	2
Cora_DS	751	6234	9
Cora_HA	400	3989	7
Cora_ML	1617	8329	7
Cora_OS	1246	6737	4
Cora_PL	1575	7949	9
WK_CL	827	4134	7
WK_TX	814	4029	7
WK_WT	1166	4165	7
WK_WC	1210	4189	7
News4	3970	8014	4

[1]http://www.cs.umass.edu/ mccallum/code-data.html
[2]http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/
[3]http://people.csail.mit.edu/jrennie/20Newsgroups/

5.3 Parameter Settings

We set the number of clusters equal to the true number of classes for all the clustering algorithms. For NCut [19], the scale parameter of Gaussian kernel for constructing adjacency matrix is set by the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. For PCA+K-means, the reduced dimension of PCA is set to the minimal number that preserves at least 95% of the information. For ADR and ASI, the dimensionality of the subspace is set to $c-1$ as in [7] [15], where c is the number of clusters. For MMC, it is implemented based on [27] [28]. For MMC and our method, the regularization parameter C is set by searching the grid $\{0.01 : 0.01 : 0.1, 0.2 : 0.1 : 1, 2 : 1 : 10\}$. ϵ is fixed to 0.1. The balance constant l is set by the grid $\{1, 5, 10\}$. Under each parameter setting, we repeat clustering 20 times, and the average result is computed. We report the best average result corresponding to the best parameter setting for each method to compare with each other.

5.4 Clustering Results

Table 2 shows the clustering accuracy of all the methods on all the data sets, while Table 3 shows the normalized mutual information.

From Table 2 and Table 3, we observe that: (1) Our method (SMMC) outperforms the other methods (including MMC) significantly on most data sets. This indicates that doing MMC in a proper subspace can improve its cluster performance; (2) Among clustering methods without dimensional reduction, MMC usually outperforms normalized cuts (NC) and K-means (KM); (3) Among dimensionality reduction based clustering methods, the methods which perform dimensionality reduction and clustering simultaneously are comparable or better than PCA+K-means (PCAKM). DisKM is often better than PCAKM, ADR and ASI. And SMMC outperforms DisKM significantly on most data sets. The reason is probably that SMMC is along the line of MMC, which is superior to K-means.

6. CONCLUSION

In this paper, we proposed an subspace maximum margin clustering (SMMC) for high dimensional data, which aims to learn a subspace, in which it tries to find a cluster assignment of the data points, together with a hyperplane classifier, such that the resultant margin is maximized among all possible cluster assignments and all possible subspaces. This problem can be solved via cutting plane and constrained concave-convex procedure (CCCP) which owns well-studied convergence properties. Furthermore, we show that the subproblem in each iteration of CCCP is joint convex, hence we can use alternating minimization to obtain the global optimum. Empirical studies illustrate that the proposed method outperforms the state of the art clustering methods.

7. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No.60721003, No.60673106 and No.60573062) and the Specialized Research Fund for the Doctoral Program of Higher Education. We thank the anonymous reviewers for their helpful comments.

8. REFERENCES

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[4] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *ICDM*, 2008.

[5] B. Chen, W. Lam, I. Tsang, and T.-L. Wong. A semi-supervised framework for feature mapping and multiclass classification. In *SDM*, pages 341–352, 2009.

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[7] C. H. Q. Ding, X. He, H. Zha, and H. D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *ICDM*, pages 147–154, 2002.

[8] C. H. Q. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and -means clustering. In *ICML*, pages 521–528, 2007.

[9] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *SDM*, 2004.

[10] F. Gieseke, T. Pahikkala, and O. Kramer. Fast evolutionary maximum margin clustering. In *ICML*, 2009.

[11] Y. Hu, J. Wang, N. Yu, and X.-S. Hua. Maximum margin clustering with pairwise constraints. In *ICDM*, pages 253–262, 2008.

[12] T. Joachims. Training linear svms in linear time. In *KDD*, pages 217–226, 2006.

[13] J. E. Kelley. The cutting plane method for solving convex programs. *Journal of the SIAM*, 8:703–712, 1960.

[14] F. D. la Torre and T. Kanade. Discriminative cluster analysis. In *ICML*, pages 241–248, 2006.

[15] T. Li, S. Ma, and M. Ogihara. Document clustering via adaptive subspace iteration. In *SIGIR*, pages 218–225, 2004.

[16] Y. Li, I. Tsang, J. Kwok, and Z. Zhou. Tighter and convex maximum margin clusteringe. In *AISTATS*, 2009.

[17] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[18] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[20] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *NIPS*, pages 1417–1424, 2006.

[21] A. S. Vishwanathan, A. J. Smola, and S. V. N. Vishwanathan. Kernel methods for missing variables. In *Proceedings of the Tenth International Workshop on AISTATS*, pages 325–332, 2005.

[22] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, 2004.

Table 2: Clustering Accuracy on the data sets

Data set	KM	NCut	MMC	PCAKM	ADR	ASI	DisKM	SMMC
ionosphere	0.7066	0.7009	0.7123	0.7123	0.7066	0.7191	0.7080	**0.7493**
Digits 3v8	0.9440	0.9384	0.9580	0.9468	0.9440	0.9496	0.9664	**0.9768**
Digits 1v7	0.9972	0.9945	1.0000	1.0000	0.9972	0.9972	0.9972	**1.0000**
Digits 2v7	0.9742	0.9298	0.9831	0.9691	0.9691	0.9860	0.9916	**1.0000**
Digits 8v9	0.8799	0.8842	0.9229	0.9124	0.9011	0.9367	0.9492	**1.0000**
Letter AvB	0.9248	0.9267	0.9421	0.8633	0.9248	0.9219	0.9273	**0.9694**
Text1	0.9440	0.9338	0.9386	0.5005	0.9053	0.7458	0.9384	**0.9530**
Text2	0.9131	0.8593	0.7189	0.5045	0.8884	0.7725	0.9290	**0.9312**
Cora_DS	0.3185	0.3423	0.3397	0.3222	0.3278	0.2955	0.3144	**0.4447**
Cora_HA	0.3777	0.3453	0.5325	0.3925	0.3385	0.3098	0.4270	**0.7025**
Cora_ML	0.4607	0.4617	0.5343	0.3596	0.3951	0.4499	0.4650	**0.6735**
Cora_OS	0.5031	0.5265	0.5778	0.5056	0.5180	0.5054	0.5992	**0.7713**
Cora_PL	0.3389	0.3195	0.4152	0.3251	0.3249	0.3083	0.3717	**0.5835**
WK_CL	0.3346	0.3481	0.6336	0.5398	0.3141	0.3430	0.5923	**0.6977**
WK_TX	0.3732	0.3468	0.4908	0.5543	0.3607	0.3416	0.5549	**0.6020**
WK_WT	0.3412	0.3489	0.4717	0.6249	0.3302	0.3325	0.7202	**0.8276**
WK_WC	0.3707	0.4124	0.5264	0.5340	0.3571	0.3507	**0.5375**	0.4777
News4	0.8325	0.7402	0.8287	0.2545	0.8364	0.6821	0.8553	**0.9582**

Table 3: Normalized Mutual Information on the data sets

Data set	KM	NCut	MMC	PCAKM	ADR	ASI	DisKM	SMMC
ionosphere	0.1212	0.1103	0.1349	0.1349	0.1193	0.1399	0.1226	**0.2602**
Digits 3v8	0.7089	0.6717	0.7659	0.7258	0.7089	0.7448	0.8204	**0.8458**
Digits 1v7	0.9752	0.9560	1.0000	1.0000	0.9752	0.9752	0.9752	**1.0000**
Digits 2v7	0.8400	0.6711	0.8824	0.8178	0.8178	0.9073	0.9382	**1.0000**
Digits 8v9	0.5461	0.4848	0.6122	0.5717	0.5349	0.6662	0.6077	**1.0000**
Letter AvB	0.6562	0.6895	0.7270	0.4564	0.6562	0.6414	0.6732	**0.7643**
Text1	0.6974	0.6489	0.6799	0.0064	0.5932	0.3385	0.6710	**0.7835**
Text2	0.6555	0.5147	0.2718	0.0190	0.5807	0.3729	0.6742	**0.6916**
Cora_DS	**0.2106**	0.2073	0.1860	0.1687	0.1975	0.1814	0.1868	0.1863
Cora_HA	0.2400	0.2127	0.2856	0.1941	0.2098	0.2173	0.2487	**0.5382**
Cora_ML	0.2685	0.2469	0.3220	0.1772	0.2205	0.2486	0.2706	**0.4449**
Cora_OS	0.1706	0.1698	0.1987	0.1328	0.1937	0.1900	0.2468	**0.4375**
Cora_PL	0.2034	0.1799	0.1675	0.1723	0.1841	0.1595	0.2713	**0.3359**
WK_CL	0.1906	0.2120	0.2664	0.1232	0.2058	0.2376	0.2286	**0.3262**
WK_TX	0.2295	0.2317	0.1360	0.1872	0.2308	0.2061	0.2356	**0.2492**
WK_WT	0.2348	0.2245	0.1653	0.1747	0.2257	0.2086	0.3547	**0.5292**
WK_WC	0.2083	0.2235	0.1095	0.1307	0.2103	0.1715	**0.3229**	0.2208
News4	0.7118	0.6238	0.6981	0.0311	0.6770	0.5555	0.7042	**0.8450**

[23] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, pages 904–910, 2005.

[24] J. Ye, Z. Zhao, and H. Liu. Adaptive distance metric learning for clustering. In *CVPR*, 2007.

[25] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, 2007.

[26] K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. In *ICML*, pages 1119–1126, 2007.

[27] B. Zhao, F. Wang, and C. Zhang. Efficient maximum margin clustering via cutting plane algorithm. In *SDM*, pages 751–762, 2008.

[28] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *ICML*, pages 1248–1255, 2008.

APPENDIX

A. LEMMA

Lemma A.1 *For any* $\mathbf{b} = [b_1, \ldots, b_n] \in \mathbb{R}^d$*, we have*

$$\min_{\lambda_i>0, \sum_{i=1}^d \lambda_i \leq 1} \sum_{i=1}^{d} \frac{b_i^2}{\lambda_i} = ||\mathbf{b}||_1^2$$

and the optima is $\lambda_i^* = \frac{|b_i|}{||\mathbf{b}||_1}$.

Proof. The proof is based on [1]. According to Cauchy-Schwarz inequality, we have $||\mathbf{b}||_1 = \sum_{i=1}^d \lambda_i^{\frac{1}{2}} \lambda_i^{-\frac{1}{2}} |b_i| \leq (\sum_{i=1}^d \lambda_i)^{\frac{1}{2}} (\sum_{i=1}^d \lambda_i^{-1} b_i^2)^{\frac{1}{2}} \leq (\sum_{i=1}^d \lambda_i^{-1} b_i^2)^{\frac{1}{2}}$ with equality if and only if $\sum_{i=1}^d \lambda_i = 1$, and $\frac{|b_i|}{\lambda_i} = \frac{|b_j|}{\lambda_j} = k$. Hence the optima is $\lambda_i^* = \frac{|b_i|}{||\mathbf{b}||_1}$ □